

SISTEMAS BASADOS EN MICROPROCESADOR
2º Grado Ingeniería Informática
Escuela Politécnica Superior - UAM

JULIO 2013

EXAMEN FINAL EXTRAORDINARIO

PREGUNTAS DEL PROBLEMA - SOLUCIÓN

NOMBRE : _____ **DNI :** _____

APELLIDOS : _____

P1. Implemente el código en C de un sencilla aplicación de ejemplo que muestre cómo utilizar las funciones de la librería para transmitir datos (caracteres) introducidos desde el teclado. El programa debe terminar cuando pulsemos la tecla ESC (Suponga que al pulsar esa tecla se leerá 27h del buffer del teclado). Se valorará la sencillez del código, así como los comentarios que aporten claridad al mismo. (3 p).

Programa Principal

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

/* Prototipos de las funciones escritas en ensamblador */
extern int far DetectarDriver ();
extern void far DesinstalarDriver ();
extern int far TransmitirDato (char);

void main(void)
{
    char dato;
    if (DetectarDriver()==1)
    {
        printf("Driver no instalado.\n");
        exit(0);
    }
    dato = getc();
    while (dato != 27h)
    {
        if (TransmitirDato(dato) ==1)
        {
            printf("Error en la transmisión del dato al módulo de
            comunicaciones.\n");
        }
        dato = getc();
    }
    DesinstalarDriver();
    printf("Fin de la transmisión.\n");
    exit(0);
} /* Fin del Programa Principal */
```

P2. Implemente el código de las funciones de la librería en ensamblador, de forma que podamos crear una librería equivalente que debe poder funcionar con el programa principal desarrollado en C de la pregunta P1 sin hacer cambios. Tenga en cuenta que durante la detección del driver no deberá llamarse al mismo a menos que se compruebe que el vector de interrupción está instalado. Se valorará la sencillez y claridad del código. Incluya comentarios que hagan más fácil su mantenimiento. (3 p.)

Funciones de la librería en ensamblador

```
_DetectarDriver proc far
    push es
    xor ax,ax
    mov es,ax
    cmp word ptr es:[60h*4],0
    jne _detectar_int
    cmp word ptr es:[60h*4+2],0
    je _detectar_nodriver
_detectar_int:
    mov ah,00h
    int 60h
    cmp ax,F0F0h
    jne _detectar_nodriver
    xor ax,ax
    jmp _detectar_fin
_detectar_nodriver:
    mov ax,1
_detectar_fin:
    pop es
    ret
_DetectarDriver endp

_DesinstalarDriver proc far
    mov ah,01h
    int 60h
    ret
_DesinstalarDriver endp

_TransmitirDato proc far
    push bp
    push cx
    mov bp,sp
    mov cx,bp[8]
    mov ax,cx ; En AL queda el dato a transmitir
    mov ah,02h
    int 60h
    mov ah,00h
    pop cx
    pop bp
    ret
_TransmitirDato endp
```

P3. Tras analizar el código del *driver* incluido como parte del enunciado responda a las siguientes cuestiones. Sea claro en sus respuesta. (4 p.).

P3.1 ¿Qué función hace el RTC en el *driver*? Indique el número de interrupciones por segundo que genera (aproximadamente) en el programa y cómo son utilizadas por el *driver* para establecer el mecanismo o función al que hace referencia la pregunta. (2 p.)

El RTC está programado para generar unas 1000 interrupciones por segundo. La rutina de servicio asociada a las interrupciones del RTC mide si el tiempo transcurrido desde que se envió el dato al módulo de comunicaciones ha alcanzado los 2 segundos, es decir, han transcurrido 2000 interrupciones. Este mecanismo se llama "time-out" y su función es evitar que el *driver* se quede en un bucle de espera infinito si no llega la confirmación por parte del módulo de comunicaciones a través de la señal "ACK" del LPT1.

P3.2 ¿Cómo indica el *driver* si está o no instalado (residiendo en memoria)? Sea claro en su respuesta. (1 p).

Cuando se llama a la función 0h del *driver* mediante la interrupción 60h, si el *driver* está instalado, éste devuelve en AX un valor especial (F0F0h) que sólo es posible si el *driver* está en ejecución.

P3.3 Indique qué bits (pines) del LPT1 se utilizan como señales de protocolo hardware (registro y nombre del bit o pin). Sea claro en su respuesta. (1 p.)

Como señal (salida) de indicación de nuevo dato para transmitir en el puerto de datos se utiliza el bit #INIT del registro de control que se encuentra a "1" en reposo, y como señal (entrada) de indicación de dato transmitido sin problemas y solicitud de nuevo dato se utiliza el bit #ACK del registro de estado del LPT1.